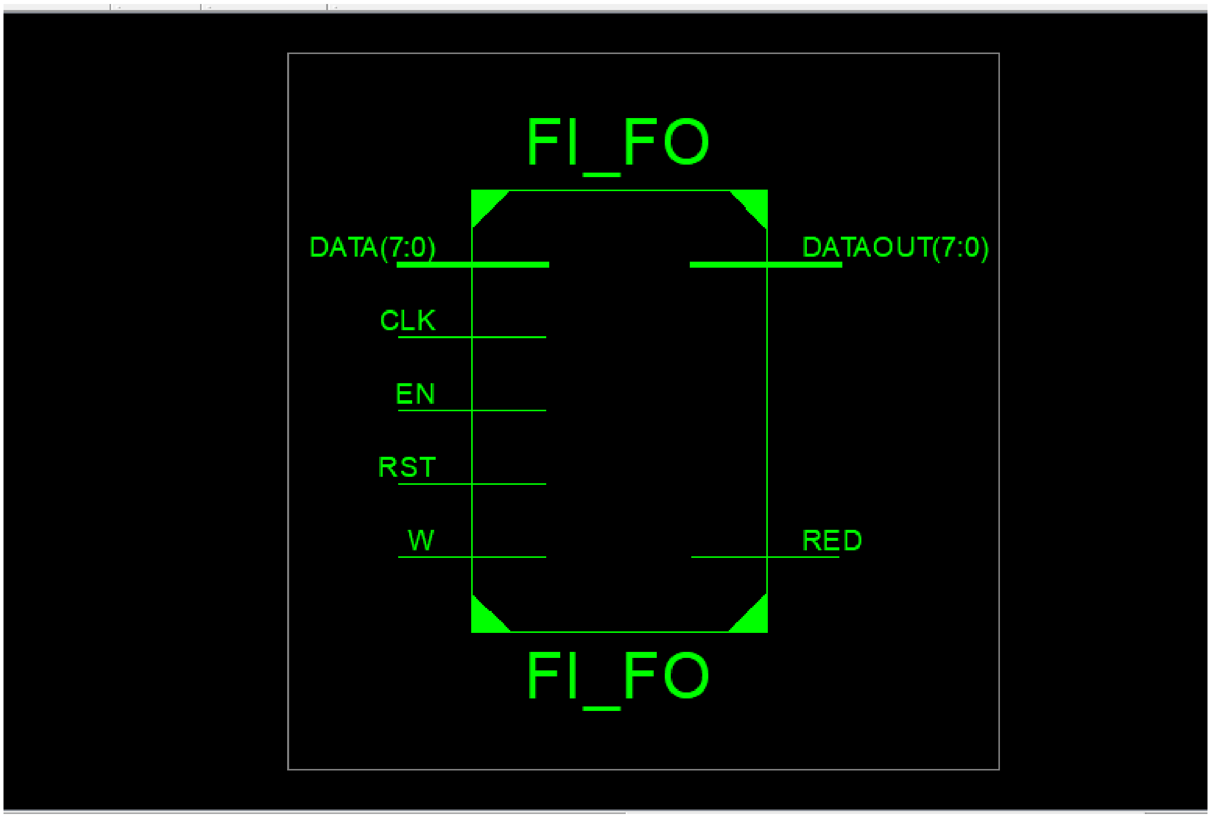
FIFO

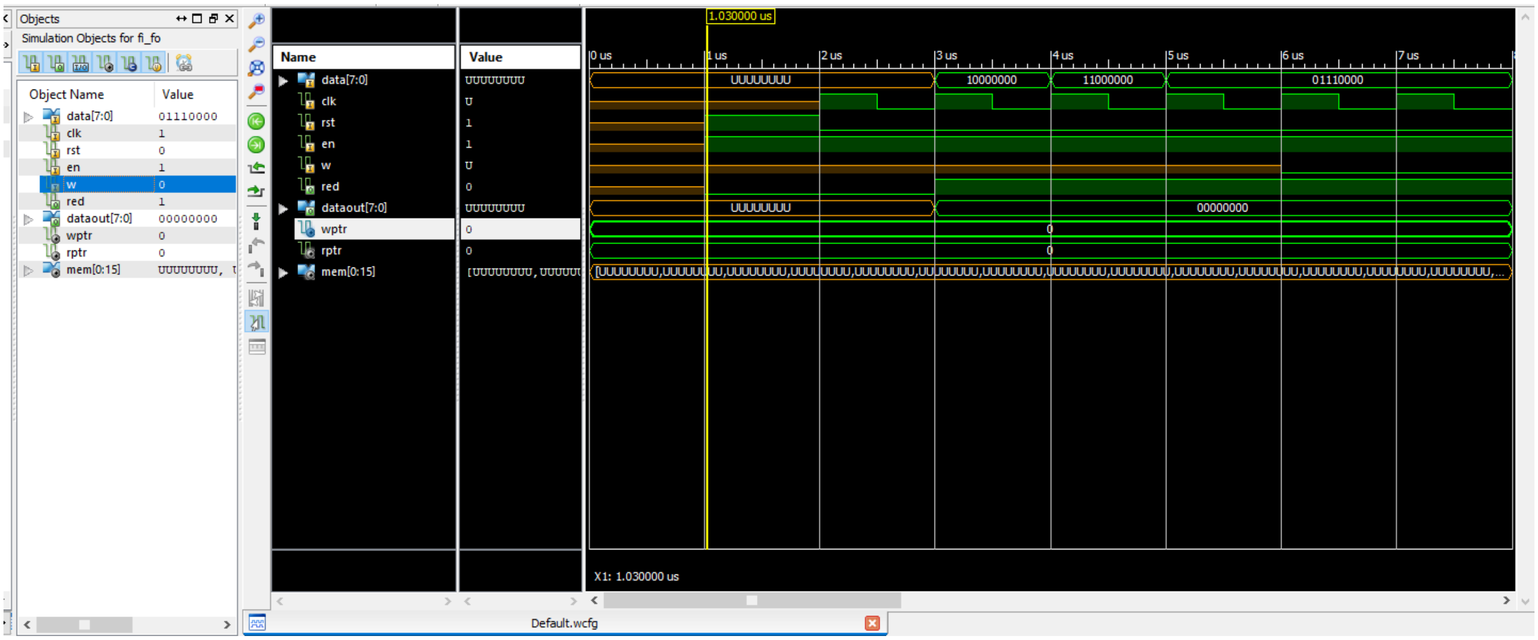
Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity FI_FO is
    Port (
        DATA : in STD_LOGIC_VECTOR(7 downto 0);
        CLK : in STD_LOGIC;
        RST : in STD_LOGIC;
        EN : in STD_LOGIC;
        W : in STD_LOGIC;
        RED : out STD_LOGIC;
        DATAOUT : out STD_LOGIC_VECTOR(7 downto 0)
    );
end FI_FO;

architecture Behavioral of FI_FO is
    signal WPTR, RPTR : integer range 0 to 15 := 0;
    type FIFO is array (0 to 15) of STD_LOGIC_VECTOR(7 downto 0);
    signal MEM : FIFO;
begin
    process (CLK, RST)
    begin
        if RST = '1' then
            WPTR <= 0;
            RPTR <= 0;
            RED <= '0';
        elsif rising_edge(CLK) then
            if EN = '1' then
                if W = '1' then
                    if WPTR < 15 then
                        MEM(WPTR) <= DATA;
                        WPTR <= WPTR + 1;
                    else
                        RED <= '1';
                    end if;
                else
                    if RPTR < WPTR then
                        RED <= '0';
                        DATAOUT <= MEM(RPTR);
                        RPTR <= RPTR + 1;
                    else
                        RED <= '1';
                        DATAOUT <= "00000000";
                    end if;
                end if;
            end if;
        end if;
    end process;
end Behavioral;
```

RTL Schematic:



Output:



Object Name	Value
data[7:0]	01110000
clk	1
rst	0
en	1
w	0
red	1
dataout[7:0]	00000000
wptr	0
rptr	0
mem[0:15]	UUUUUUUU, U

Name	Value
data[7:0]	UUUUUUUU
clk	U
rst	1
en	1
w	U
red	0
dataout[7:0]	UUUUUUUU
wptr	0
rptr	0
mem[0:15]	[UUUUUUUU, UUUUUU